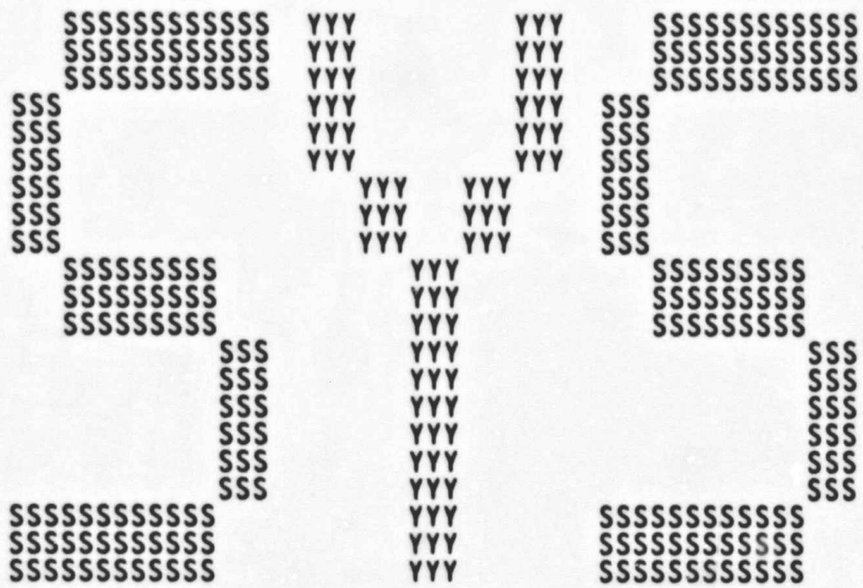
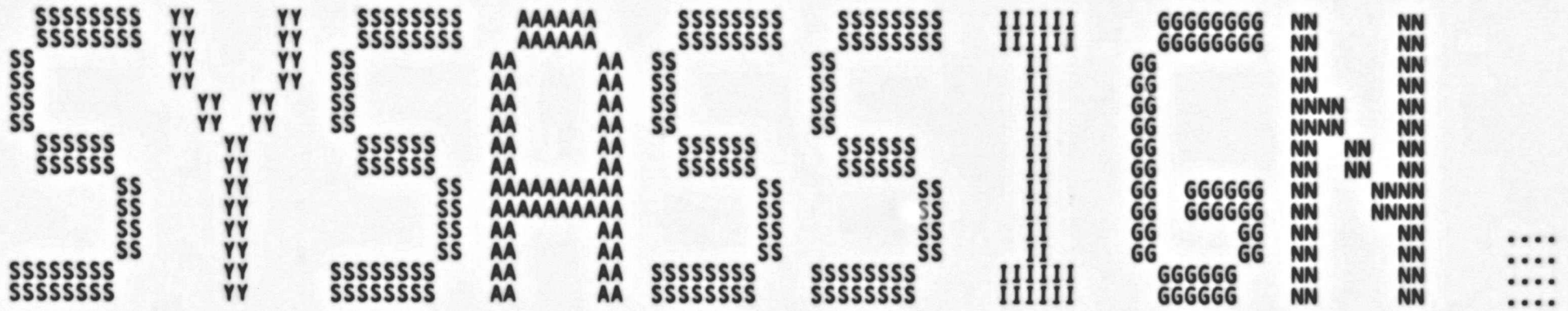

SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS

FILEID**SYSASSIGN



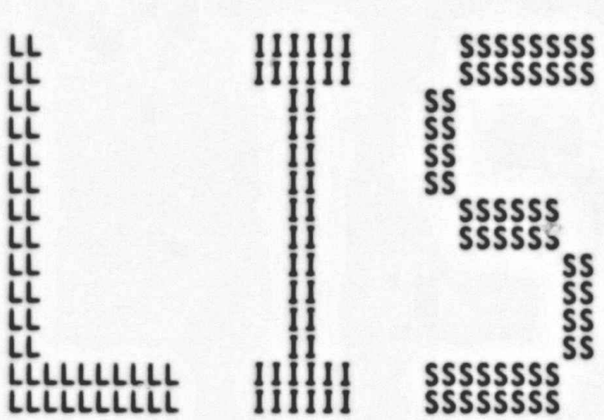


Table of contents

```
0000     1          .TITLE  SYSASSIGN - SYSTEM SERVICE ASSIGN I/O CHANNEL
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;****************************************************************************
0000     6  ;*                                                                          *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                    *
0000    10  ;*                                                                          *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16  ;*  TRANSFERRED.                                                            *
0000    17  ;*                                                                          *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20  ;*  CORPORATION.                                                            *
0000    21  ;*                                                                          *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    24  ;*                                                                          *
0000    25  ;*                                                                          *
0000    26  ;****************************************************************************
0000    27  ;
0000    28  ; D. N. CUTLER 25-AUG-76
0000    29  ;
0000    30  ; SYSTEM SERVICE ASSIGN I/O CHANNEL
0000    31  ;
0000    32  ; MODIFIED BY:
0000    33  ;
0000    34  ;       V03-025 LMP0274         L. Mark Pilant          11-Jul-1984  9:27
0000    35  ;               Fix a bug introduced in LMP0221 that caused read access to be
0000    36  ;               necessary to assign a channel to a shared device.
0000    37  ;
0000    38  ;       V03-024 KPL0003         Peter Lieberwirth       3-May-1984
0000    39  ;               Fix problem with remote channel assignment introduced
0000    40  ;               in V03-022.
0000    41  ;
0000    42  ;       V03-023 TMK0001         Todd M. Katz            28-Apr-1984
0000    43  ;               Eliminate the $LOGDEF data definitions.
0000    44  ;
0000    45  ;       V03-022 KPL0002         Peter Lieberwirth       24-Apr-1984
0000    46  ;               Fix problems with remote device assignment introduced in
0000    47  ;               V03-020.
0000    48  ;
0000    49  ;       V03-021 RKS0021         RICK SPITZ              10-APR-1984
0000    50  ;               Fix problem in assign with shadow set unit.
0000    51  ;               Add support for physical terminal UCB redirection to
0000    52  ;               a logical UCB when DEV$V_RED is set in DEVCHAR2
0000    53  ;
0000    54  ;       V03-020 KPL0001         Peter Lieberwirth       9-Apr-1984
0000    55  ;               1. If the high bit in the ACMODE byte is set, don't
0000    56  ;                  translate the logical name, because RMS already did.
C000    57  ;
```

```
0000     58 ;                2. Use LNM$SEARCH_ONE to translate the remote device name.
0000     59 ;                   Allocate a KRP to contain the equivlence string since 255
0000     60 ;                   bytes is too much kernel stack to use.  Recursively
0000     61 ;                   translate the logical name.
0000     62 ;
0000     63 ;                3. Use LNM$C_MAXDEPTH as the maximum logical name recursion
0000     64 ;                   depth.
0000     65 ;
0000     66 ;       V03-019 LMP0221         L. Mark Pilant,         30-Mar-1984  15:38
0000     67 ;               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
0000     68 ;               ORB$W_PROT.
0000     69 ;
0000     70 ;       V03-018 ACG0399         Andrew C. Goldstein,    24-Feb-1984  21:42
0000     71 ;               Track I/O database search and interlock rewrite;
0000     72 ;               remove generic assign feature
0000     73 ;
0000     74 ;       V03-017 EMD0045         Ellen M. Dusseault      1-Feb-1984
0000     75 ;               Add check for physical io privilege (phy_io) if device
0000     76 ;               is a shadow set member.
0000     77 ;
0000     78 ;       V03-016 LMP0185         L. Mark Pilant,         1-Feb-1984  13:49
0000     79 ;               Add support for device ACLs.
0000     80 ;
0000     81 ;       V03-015 TCM0006         Trudy C. Matthews       18-Jan-1984
0000     82 ;               Report SS$_NOTQUEUED status from $ENQ as SS$_DEVALLOC.
0000     83 ;               NOTQUEUED means that the device is allocated elsewhere in
0000     84 ;               the cluster.
0000     85 ;
0000     86 ;       V03-014 TCM0005         Trudy C. Matthews       7-Oct-1983
0000     87 ;               Only take out a lock on the device if the system is
0000     88 ;               currently actively participating in a cluster.
0000     89 ;
0000     90 ;       V03-013 TCM0004         Trudy C. Matthews       12-Sep-1983
0000     91 ;               Only take out a lock on the device if the system is a member
0000     92 ;               of a cluster.
0000     93 ;
0000     94 ;       V03-012 TCM0003         Trudy C. Matthews       16-Jun-1983
0000     95 ;               Return status from EXE$LOCK_DEV rather than overwriting it
0000     96 ;               with SS$_DEVALLOC when we fail to obtain the lock.  Also
0000     97 ;               use input register R1 to signal EXE$LOCK_DEV that we're
0000     98 ;               not interested in the lock value block.  Add ability to
0000     99 ;               request a generic device channel.  Change lock mode from
0000    100 ;               PR to CR.
0000    101 ;
0000    102 ;       V03-011 TCM0002         Trudy C. Matthews       26-May-1983
0000    103 ;               Allocate the UCB on the local system while taking out the
0000    104 ;               cluster-wide lock.  This is to disallow changes to the UCB
0000    105 ;               while the locking code executes (at IPL 0, and without the
0000    106 ;               I/O database mutex).
0000    107 ;
0000    108 ;       V03-010 TCM0001         Trudy C. Matthews       13-May-1983
0000    109 ;               If this is the first $ASSIGN to a cluster-wide device,
0000    110 ;               take out a cluster-wide lock showing that this device has active
0000    111 ;               channels.
0000    112 ;
0000    113 ;       V03-009 JLV0240         Jake VanNoy             11-APR-1983
0000    114 ;               Prevent user with SHARE privilege from becoming owner
```

```
0000    115 ;                   of an already owned device.
0000    116 ;
0000    117 ;       V03-008 ROW0165         Ralph O. Weber          25-FEB-1983
0000    118 ;                   Fix cloned UCB logic to debit BYTCNT before calling the
0000    119 ;                   driver's CLONEDUCB routine, and to credit BYTCNT if the
0000    120 ;                   CLONEDUCB routine vetos the cloning.  Change cloned UCB logic
0000    121 ;                   to not set put PCB$L_PID in UCB$L_PID if the DEV$M_SHR bit is
0000    122 ;                   set in UCB$L_DEVCHAR of the cloned UCB.
0000    123 ;
0000    124 ;       V03-007 JLV0230         Jake VanNoy             24-FEB-1983
0000    125 ;                   Add use of new SHARE privilege to allow assignment of
0000    126 ;                   channel to an allocated non-sharable device.
0000    127 ;
0000    128 ;       V03-006 DMW4009         DMWalp                  17-Nov-1982
0000    129 ;                   Recoded call internal call to $TRNLOG to be external.
0000    130 ;
0000    131 ;       V03-005 ROW0138         Ralph O. Weber           8-NOV-1982
0000    132 ;                   Add to UCB cloning a check for mailbox device characteristic
0000    133 ;                   with automatic setting of device status bit UCB$V_DELMBX when
0000    134 ;                   DEV$V_MBX is set in UCB$L_DEVCHAR.  This duplicates in source
0000    135 ;                   the patch made to the last two system images.  The source
0000    136 ;                   change is being made to allow NETDRIVER to track V3.x releases
0000    137 ;                   and still work on the base level systems.  Once this need is
0000    138 ;                   no longer present, this device dependent function can be
0000    139 ;                   removed.
0000    140 ;
0000    141 ;       V03-004 ROW0132         Ralph O. Weber          13-OCT-1982
0000    142 ;                   Correct call to driver's CLONEDUCB routine to conform with
0000    143 ;                   specification.
0000    144 ;
0000    145 ;       V03-003 ROW0127         Ralph O. Weber           4-OCT-1982
0000    146 ;                   Make changes required to use new UCB creation routines in
0000    147 ;                   UCBCREDEL.  Change netork assignment to cloning assignment
0000    148 ;                   with test of NETMBX priviledge iff DEV$M_NET is set in
0000    149 ;                   UCB$L_DEVCHAR.  Rewrite and modernize cloning assignment.
0000    150 ;                   Eliminate second call to TEST_MAILBOX in cloning assignment
0000    151 ;                   code path since all that is really desired the R6 result of
0000    152 ;                   the previous call and R6 is preserved by the cloning
0000    153 ;                   assignment code.
0000    154 ;
0000    155 ;       V03-002 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    156 ;                   Added $DEVDEF and fixed comments.
0000    157 ;
0000    158 ;       V03-001 PHL0101         Peter H. Lipman 21-Jun-1982
0000    159 ;                   $QIOW now synchronizes the EFN and IOSB parameters
0000    160 ;                   correctly.  Eliminate the synchronization code here.
0000    161 ;
0000    162 ;
0000    163 ;--
0000    164
0000    165 ;
0000    166 ; MACRO LIBRARY CALLS
0000    167 ;
0000    168
0000    169         $CCBDEF                         ;DEFINE CCB OFFSETS
0000    170         $CLUBDEF                        ;DEFINE CLUSTER BLOCK OFFSETS
0000    171         $CRBDEF                         ;DEFINE CONTROLLER BLOCK OFFSETS
```

```
                        0000   172          $DDTDEF                         ;DRIVER DISPATCH TABLE
                        0000   173          $DEVDEF                         ;DEFINE DEVICE TYPE CODES
                        0000   174          $IODEF                          ;DEFINE I/O FUNCTION CODES
                        0000   175          $IOCDEF                         ;DEFINE IOC FLAG BITS
                        0000   176          $JIBDEF                         ;DEFINE JIB OFFSETS
                        0000   177          $LCKDEF                         ;DEFINE LOCK MANAGER SYMBOLS
                        0000   178          $LNMDEF                         ;DEFINE LOGICAL NAME BLOCK OFFSETS
                        0000   179          $LNMSTRDEF                      ;DEFINE LOGICAL NAME BLOCK OFFSETS
                        0000   180          $ORBDEF                         ;DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
                        0000   181          $PCBDEF                         ;DEFINE PCB OFFSETS
                        0000   182          $PRDEF                          ;DEFINE PROCESSOR REGISTERS
                        0000   183          $PRVDEF                         ;DEFINE PRIVILEGE BITS
                        0000   184          $SSDEF                          ;DEFINE SYSTEM STATUS VALUES
                        0000   185          $TTYUCBDEF                      ;DEFINE TERMINAL SPECIFIC UCB FIELDS
                        0000   186          $UCBDEF                         ;DEFINE UCB OFFSETS
                        0000   187          $VECDEF                         ;DEFINE VECTOR OFFSETS
                        0000   188
                        0000   189 ;
                        0000   190 ; LOCAL SYMBOLS
                        0000   191 ;
                        0000   192 ; ARGUMENT LIST OFFSET DEFINITIONS
                        0000   193 ;
                        0000   194
                00000004 0000  195 DEVNAM=4                                ;ADDRESS OF DEVICE NAME STRING DESCRIPTOR
                00000008 0000  196 CHAN=8                                  ;ADDRESS TO STORE ASSIGNED CHANNEL NUMBER
                0000000C 0000  197 ACMODE=12                               ;ACCESS MODE
                00000010 0000  198 MBXNAM=16                               ;ADDRESS OF MAILBOX NAME STRING DESCRIPTOR
                        0000   199
                        0000   200 ;
                        0000   201 ; STACK LOCAL STORAGE SYMBOL
                        0000   202 ;
                        0000   203          $OFFSET 0,NEGATIVE,<-
                        0000   204                  MAXACMODE>                ;MAXIMIZED ACCESS MODE
                        FFFC   205 MAXACMODE:
                        0000   206
                        0000   207 ;
                        0000   208 ; LOCAL DATA
                        0000   209 ;
                        0000   210
                00000000 0000  211          .PSECT  Y$EXEPAGED
                        0000   212
                        0000   213 LNM_TBL:
49 46 24 4D 4E 4C 00000008'010E0000' 0000  214          .ASCID  "LNM$FILE_DEV"    ;LOGICAL NAME TABLE FOR DEVICES
                56 45 44 5F 45 4C  000E
                        0014   215
                   54 45 4E 5F  0014  216 NETNAM: .ASCII  /_NET/                ;NETWORK DEVICE NAMe
                        0018   217 NETEND:                                    ;REFERENCE LABEL
```

```
0018   219          .SBTTL  ASSIGN I/O CHANNEL
0018   220 ;+
0018   221 ; EXE$ASSIGN - ASSIGN I/O CHANNEL
0018   222 ;
0018   223 ; THIS SERVICE PROVIDES THE CAPABILITY TO ASSIGN A DEVICE TO AN I/O CHANNEL
0018   224 ; AND ESTABLISH NECESSARY DEVICE LINKAGE AND CONTROL INFORMATION IN THE
0018   225 ; ASSOCIATED CHANNEL CONTROL BLOCK. OPTIONALLY A MAILBOX CAN ALSO BE
0018   226 ; SPECIFIED WHICH WILL RECEIVE UNSOLICITED INPUT SENT TO THE ASSIGNED
0018   227 ; DEVICE.
0018   228 ;
0018   229 ; INPUTS:
0018   230 ;
0018   231 ;       DEVNAM(AP) = ADDRESS OF DEVICE NAME STRING DESCRIPTOR.
0018   232 ;       CHAN(AP)   = ADDRESS TO STORE ASSIGNED CHANNEL NUMBER.
0018   233 ;       ACMODE(AP) = ACCESS MODE CHANNEL IS TO BE ASSIGNED TO.
0018   234 ;                    HIGH BIT OF ACMODE BYTE SET MEANS DON'T TRANSLATE
0018   235 ;                    LOGICAL NAME.
0018   236 ;       MBXNAM(AP) = ADDRESS OF MAILBOX NAME STRING DESCRIPTOR (ZERO IMPLIES
0018   237 ;                    NONE).
0018   238 ;
0018   239 ;       R4 = CURRENT PROCESS PCB ADDRESS.
0018   240 ;
0018   241 ; OUTPUTS:
0018   242 ;
0018   243 ;       R0 LOW BIT CLEAR INDICATES FAILURE TO ASSIGN CHANNEL TO DEVICE.
0018   244 ;
0018   245 ;               R0 = SS$_ACCVIO - DEVICE NAME STRING, DEVICE NAME STRING
0018   246 ;                    DESCRIPTOR, MAILBOX NAME STRING, OR MAILBOX NAME
0018   247 ;                    STRING DESCRIPTOR CANNOT BE READ BY CALLING ACCESS
0018   248 ;                    MODE, OR CHANNEL NUMBER CANNOT BE WRITTEN BY CALLING
0018   249 ;                    ACCESS MODE.
0018   250 ;
0018   251 ;               R0 = SS$_DEVALLOC - DEVICE ALLOCATED TO ANOTHER PROCESS.
0018   252 ;
0018   253 ;               R0 = SS$_DEVNOTMBX - SPECIFIED MAILBOX DEVICE IS NOT A
0018   254 ;                    MAILBOX.
0018   255 ;
0018   256 ;               R0 = SS$_EXQUOTA - PROCESS HAS INSUFFICIENT BUFFER QUOTA TO
0018   257 ;                    ALLOCATE NETWORK UCB.
0018   258 ;
0018   259 ;               R0 = SS$_INSFMEM - SUFFICIENT SYSTEM DYNAMIC MEMORY DOES NOT
0018   260 ;                    EXIST TO ALLOCATE NETWORK UCB.
0018   261 ;
0018   262 ;               R0 = SS$_IVDEVNAM - DEVICE OR MAILBOX NAME STRING CONTAINS
0018   263 ;                    INVALID CHARACTERS, OR NO DEVICE NAME STRING
0018   264 ;                    DESCRIPTOR SPECIFIED.
0018   265 ;
0018   266 ;               R0 = SS$_IVLOGNAM - ZERO OR GREATER THAN MAXIMUM LENGTH
0018   267 ;                    DEVICE OR MAILBOX NAME STRING SPECIFIED.
0018   268 ;
0018   269 ;               R0 = SS$_TOOMANYLNAM - ITERATION LIMIT ON LOGICAL NAME
0018   270 ;                    TRANSLATION EXCEEDED.
0018   271 ;
0018   272 ;               R0 = SS$_NOIOCHAN - NO I/O CHANNEL IS AVAILABLE FOR ASSIGNMENT.
0018   273 ;
0018   274 ;               R0 = SS$_NOPRIV - PROCESS DOES NOT HAVE PRIVILEGE TO CREATE
0018   275 ;                    NETWORK UCB OR DOES NOT HAVE PRIVILEGE TO ALLOCATE
```

```
                                0018   276 ;                       THE DEVICE.
                                0018   277 ;
                                0018   278 ;               R0 = SS$_NOSUCHDEV - SPECIFIED DEVICE OR MAILBOX DOES NOT
                                0018   279 ;                    EXIST ON HOST SYSTEM.
                                0018   280 ;
                                0018   281 ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                                0018   282 ;
                                0018   283 ;               R0 = SS$_REMOTE - NORMAL COMPLETION, ASSIGNMENT COMPLETED
                                0018   284 ;                    ON REMOTE SYSTEM.
                                0018   285 ;
                                0018   286 ;               R0 = SS$_NORMAL - NORMAL COMPLETION, ASSIGNMENT COMPLETED
                                0018   287 ;                    ON HOST SYSTEM.
                                0018   288 ;
                                0018   289 ;               R0 = SS$_DEVACTIVE - MAILBOX ALREADY ASSOCIATED WITH DEVICE
                                0018   290 ;-
                                0018   291
                          0FFC  0018   292          .ENTRY  EXE$ASSIGN,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                       7E   D4  001A   293          CLRL    -(SP)                   ;SPACE FOR STACK LOCAL
                    5B 08 AC   D0  001C   294          MOVL    CHAN(AP),R11            ;GET ADDRESS TO STORE CHANNEL NUMBER
                               0020   295          IFNOWRT #2,(R11),30$            ;CAN CHANNEL NUMBER BE WRITTEN?
                       6B   B4  0026   296          CLRW    (R11)                   ;CLEAR CHANNEL NUMBER IN CASE OF ERROR
                    5A 10 AC   D0  0028   297          MOVL    MBXNAM(AP),R10          ;GET ADDRESS OF MAILBOX NAME DESCRIPTOR
                       0C   13  002C   298          BEQL    10$                     ;IF EQL NO MAILBOX SPECIFIED
                               002E   299          IFNORD  #8,(R10),30$            ;CAN MAILBOX DESCRIPTOR BE READ?
                    7E 6A   7D  0034   300          MOVQ    (R10),-(SP)             ;COPY MAILBOX NAME DESCRIPTOR
                    5A 5E   D0  0037   301          MOVL    SP,R10                  ;SET ADDRESS OF MAILBOX NAME DESCRIPTOR
                 50 0144 8F   3C  003A   302 10$:     MOVZWL  #SS$_IVDEVNAM,R0        ;SET INVALID DEVICE NAME STATUS
                 59 04 AC   D0  003F   303          MOVL    DEVNAM(AP),R9           ;GET ADDRESS OF DEVICE NAME DESCRIPTOR
                       19   13  0043   304          BEQL    20$                     ;IF EQL NO DEVICE SPECIFIED
                               0045   305          IFNORD  #8,(R9),30$             ;CAN DEVICE NAME DESCRIPTOR BE READ?
50 0C AC 02   00   EF  004B   306          EXTZV   #0,#2,ACMODE(AP),R0     ;GET SPECIFIED ACCESS MODE
                 FFAC' 30  0051   307          BSBW    EXE$MAXACMODE           ;MAXIMIZE ACCESS MODE
                 FC AD   50  D0  0054   308          MOVL    R0,MAXACMODE(FP)        ;SAVE MAXIMIZED ACCESS MODE
                    FFA5' 30  0058   309          BSBW    IOC$FFCHAN              ;FIND FREE I/O CHANNEL
                    08 50  E8  005B   310          BLBS    R0,FREECHAN             ;IF LBS FREE I/O CHANNEL FOUND
                           04  005E   311 20$:     RET                             ;
                 50   0C   3C  005F   312 30$:     MOVZWL  #SS$_ACCVIO,R0          ;SET ACCESS VIOLATION STATUS
                           04  0062   313          RET                             ;
                               0063   314
                               0063   315 ;
                               0063   316 ; IF THE CALLER SETS THE HIGH BIT IN THE ACMODE BYTE, IT IS INTERPRETED HERE
                               0063   317 ; AS A FLAG INDICATING IT IS UNNECESSARY TO TRANSLATE THE LOGICAL NAME BECAUSE
                               0063   318 ; THE CALLER ALREADY HAS.
                               0063   319 ;
                               0063   320
                               0063   321          .ENABL  LSB
                    010C   31  0063   322 0$:      BRW     90$                     ;UNLOCK DATABASE AND RETURN BRANCH AID
                               0066   323 FREECHAN:                               ;FREE CHANNEL FOUND
                 57   51   7D  0066   324          MOVQ    R1,R7                   ;SAVE CHANNEL AND CCB ADDRESS
                    FF94' 30  0069   325          BSBW    SCH$IOLOCKW             ;LOCK I/O DATA BASE FOR WRITE ACCESS
                    0274  30  006C   326          BSBW    TEST_MAILBOX            ;TEST IF MAILBOX SPECIFIED
                 F1 50  E9  006F   327          BLBC    R0,0$                   ;IF LBC SEARCH FAILURE
52 41 8F   9A  0072   328          MOVZBL  #IOC$M_PHY!IOC$M_ANY,R2 ;PHYSICAL DEVICE, NO CHECKS, NO LNM MODE
                 0C AC   95  0076   329          TSTB    ACMODE(AP)              ;HIGH BIT SET INDICATES NO $TRNLNM TO DO
                       04  18  0079   330          BGEQ    3$                      ;BRANCH IF MUST TRANSLATE LOGICAL NAME
00 52   09   E2  007B   331          BBSS    #IOC$V_NO_TRANS,R2,3$   ;TELL IOC$TRANDEVNAM NOT TO DO $TRNLNM
                 51   59  D0  007F   332 3$:      MOVL    R9,R1                   ;SET ADDRESS OF DEVICE NAME DESCRIPTOR
```

```
                53   D4   0082   333          CLRL    R3                              ;NO LOCK VALUE BLOCK
              FF79'  30   0084   334          BSBW    IOC$SEARCH                      ;SEARCH FOR DEVICE
           5D 50     E9   0087   335          BLBC    R0,40$                          ;IF LBC SEARCH FAILURE
05 3C A1     08     E1   008A   336          BBC     S^#DEV$V_RED,UCB$L_DEVCHAR2(R1),4$; SKIP IF NOT REDIRECTED
                    008F   337                                                          ; PHYSICAL TERMINAL UCB
 51    00C0 C1     D0   008F   338          MOVL    UCB$L_TT_LOGUCB(R1),R1          ; REDIRECT TO ASSOCIATED LOGICAL TTY UCB
                    0094   339 ;
                    0094   340 ; DEVICE FOUND
                    0094   341 ;
                    0094   342
       55   51     D0   0094   343 4$:     MOVL    R1,R5                           ;SAVE ADDRESS OF DEVICE UCB
0D 38 A5    06     E0   0097   344          BBS     S^#DEV$V_SPL,UCB$L_DEVCHAR(R5),5$ ;IF SET, SPOOLED DEVICE
0B 3C A5    06     E0   009C   345          BBS     S^#DEV$V_SSM,UCB$L_DEVCHAR2(R5),6$ ;If set, shadow set member
14 64 A5    0D     E1   00A1   346          BBC     S^#UCB$V_TEMPLATE,-             ; Branch if this assignment is not
                    00A6   347                  UCB$L_STS(R5), LOCAL            ; to a cloned device.
            00CC    31   00A6   348          BRW     CLONE_UCB                       ; Else, brach to clone the UCB.
            0087    31   00A9   349 5$:     BRW     80$                             ; spooled device
                    00AC   350 6$:
                    00AC   351          IFPRIV  PHY_IO,LOCAL                    ; Must have phy_io priv, if shadow set membe
 50    28B4 8F     3C   00B2   352          MOVZWL  #SS$_NOPHY_IO,R0                ; Exit with physical_io priv error
            00B8    31   00B7   353          BRW     90$                             ; Unlock I/O database
                    00BA   354
                    00BA   355 ; LOCAL ASSIGNMENT
                    00BA   356 ;
                    00BA   357
                    00BA   358 LOCAL:                                               ;LOCAL ASSIGNMENT
   50    2C A5     D0   00BA   359          MOVL    UCB$L_PID(R5),R0                ;GET PROCESS ID OF OWNER
            31     13   00BE   360          BEQL    50$                             ;IF EQL DEVICE NOT ALLOCATED
       51   54     D0   00C0   361          MOVL    R4,R1                           ;COPY PROCESS PCB ADDRESS
60 A1     50     D1   00C3   362 10$:     CMPL    R0,PCB$L_PID(R1)                ;PROCESS ID MATCH?
            40     13   00C7   363          BEQL    70$                             ;IF EQL YES
   51    1C A1     3C   00C9   364          MOVZWL  PCB$L_OWNER(R1),R1              ;GET CREATOR PROCESS INDEX
            0A     13   00CD   365          BEQL    20$                             ;IF EQL NO CREATOR
51    00000000'FF41    D0   00CF   366          MOVL    @L^SCH$GL_PCBVEC[R1],R1         ;GET ADDRESS OF CREATOR PCB
               EA     11   00D7   367          BRB     10$                             ;
                    00D9   368 20$:     IFPRIV  SHARE,50$                       ;BRANCH IF SHARE PRIV ENABLED
   50    0840 8F     3C   00DF   369          MOVZWL  #SS$_DEVALLOC,R0                ;SET DEVICE ALREADY ALLOCATED
            008B    31   00E4   370 30$:     BRW     90$                             ;
                    00E7   371
                    00E7   372 ;
                    00E7   373 ; DEVICE SEARCH FAILURE
                    00E7   374 ;
                    00E7   375
   50    08F0 8F     B1   00E7   376 40$:     CMPW    #SS$_NONLOCAL,R0                ;REMOTE DEVICE?
               F6     12   00EC   377          BNEQ    30$                             ;IF NEQ NO
            00E9    31   00EE   378          BRW     REMOTE                          ;
                    00F1   379
                    00F1   380 ;
                    00F1   381 ; DEVICE NOT SPOOLED OR ALLOCATED - IF IT'S ALSO NOT SHAREABLE, CHECK THAT
                    00F1   382 ; PROCESS HAS PRIVILEGE TO ALLOCATE IT
                    00F1   383 ;
                    00F1   384
13 38 A5    10     E0   00F1   385 50$:     BBS     S^#DEV$V_SHR,UCB$L_DEVCHAR(R5),70$ ;IF SET, DEVICE SHAREABLE
              FF07'  30   00F6   386          BSBW    EXE$CHKRDACCES                  ;CHECK USER'S RIGHT TO ALLOCATE DEVICE
                    00F9   387                                                          ; R4 = PCB ADDRESS
                    00F9   388                                                          ; R5 = UCB ADDRESS
         03 50      E8   00F9   389          BLBS    R0,60$                          ;CONTINUE IF SUCCESS
```

```
                                                F 15
                0073   31   00FC   390          BRW     90$                     ;IF LBC NO PRIVILEGE
                            00FF   391 ;
                            00FF   392 ; PERFORM IMPLICIT ALLOCATION IF DEVICE NOT SHARABLE.
                            00FF   393 ; NOTE!! THIS CODE ASSUMES THAT NON-SHAREABLE DEVICES ARE NOT
                            00FF   394 ; CLUSTER VISIBLE. IF THIS ASSUMPTION EVER CHANGES, SUITABLE TESTS
                            00FF   395 ; AND A CALL TO IOC$LOCK_DEV MUST BE ADDED HERE.
                            00FF   396 ;
                            00FF   397
          2C A5   D5        00FF   398 60$:    TSTL    UCB$L_PID(R5)           ;CHECK TO SEE IF OWNED
              05   12        0102   399         BNEQ    70$                     ;BRANCH IF IT IS
2C A5     60 A4   D0        0104   400         MOVL    PCB$L_PID(R4),UCB$L_PID(R5) ;SET CURRENT PROCESS AS OWNER
                            0109   401
                            0109   402 ;
                            0109   403 ; ASSOCIATE MAILBOX IF:
                            0109   404 ;
                            0109   405 ;       1. NOT FILE DEVICE
                            0109   406 ;       2. NOT SHAREABLE DEVICE
                            0109   407 ;       3. MAILBOX NOT ALREADY ASSOCIATED
                            0109   408 ;       4. MAILBOX IS SPECIFIED
                            0109   409 ;
                            0109   410
25 38 A5    0E    E0        0109   411 70$:    BBS     S^#DEV$V_FOD,UCB$L_DEVCHAR(R5),80$ ;IF SET, FILE DEVICE
20 38 A5    10    E0        010E   412         BBS     S^#DEV$V_SHR,UCB$L_DEVCHAR(R5),80$ ;IF SET, SHARED DEVICE
              56   D5        0113   413         TSTL    R6                      ;ARE WE ASSOCIATING A MBX
              1C   13        0115   414         BEQL    80$                     ;IF NOT JUST CONTINUE
          60 A5    D5        0117   415         TSTL    UCB$L_AMB(R5)           ;IS THERE ONE CURRENTLY ASSOC?
              0D   13        011A   416         BEQL    75$                     ;IF NOT ASSOC NEW ONE
       60 A5   56   D1        011C   417         CMPL    R6,UCB$L_AMB(R5)        ;TRYING TO ASSOC DIFFERENT MBX?
              11   13        0120   418         BEQL    80$                     ;IF NOT JUST CONTINUE
 50    02C4 8F    3C        0122   419         MOVZWL  #SS$_DEVACTIVE,R0       ;DON'T DO THE ASSIGN
              49   11        0127   420         BRB     90$                     ;RETURN THE ERROR
                            0129   421
       60 A5   56   D0        0129   422 75$:    MOVL    R6,UCB$L_AMB(R5)        ;SET ASSOCIATED MAILBOX UCB ADDRESS
          5C A6    B6        012D   423         INCW    UCB$W_REFC(R6)          ;INCREMENT MAILBOX UCB REFERENCE COUNT
          56   01   9A        0130   424         MOVZBL  #CCB$M_AMB,R6           ;SET ASSOCIATED MAILBOX FLAG
                            0133   425 ;
                            0133   426 ; If this is the first $ASSIGN to a device that is available cluster-wide,
                            0133   427 ; take out a lock to show that this device is active.
                            0133   428 ;
          5C A5    B5        0133   429 80$:    TSTW    UCB$W_REFC(R5)          ;IS THIS THE FIRST CHANNEL ASSIGNED?
              24   12        0136   430         BNEQ    85$                     ;BRANCH IF NOT
                            0138   431         IFNOCLSTR       85$             ;BRANCH IF WE'RE NOT IN A CLUSTER
    17 3C A5    00   E1        0140   432         BBC     #DEV$V_CLU, -           ;BRANCH IF DEVICE IS NOT AVAILABLE
                            0145   433                 UCB$L_DEVCHAR2(R5),85$  ;CLUSTER-WIDE
 50   00000000'EF   D0        0145   434         MOVL    CLU$GL_CLUB,R0          ;GET ADDRESS OF CLUSTER BLOCK
    0B 1C A0    00   E1        014C   435         BBC     #CLUB$V_CLUSTER, -      ;BRANCH IF WE HAVEN'T JOINED THE
                            0151   436                 CLUB$L_FLAGS(R0),85$    ;CLUSTER YET
          50   01   D0        0151   437 82$:    MOVL    #LCK$K_CRMODE,R0        ;CR MODE FOR CHANNEL ASSIGNS
              51   D4        0154   438         CLRL    R1                      ;DON'T WANT VALUE BLOCK RETURNED
            FEA7'  30        0156   439         BSBW    IOC$LOCK_DEV            ;TAKE OUT A LOCK ON THE DEVICE
          16 50    E9        0159   440 83$:    BLBC    R0,90$                  ;BRANCH IF WE DIDN'T GET THE LOCK
                            015C   441
          68   55   D0        015C   442 85$:    MOVL    R5,CCB$L_UCB(R8)        ;STORE UCB ADDRESS IN CCB
             5C A5    B6        015F   443         INCW    UCB$W_REFC(R5)          ;INCREMENT UCB REFERENCE COUNT
09 A8   FC AD    01   81        0162   444         ADDB3   #1,MAXACMODE(FP),CCB$B_AMOD(R8) ;STORE ACCESS MODE OF CHANNEL
          08 A8   56   90        0168   445         MOVB    R6,CCB$B_STS(R8)        ;SET CHANNEL STATUS FLAGS
             6B   57   B0        016C   446         MOVW    R7,(R11)                ;STORE ASSIGNED CHANNEL NUMBER
```

```
                50   01   3C  016F   447                 MOVZWL  #SS$_NORMAL,R0          ;SET NORMAL COMPLETION STATUS
                   FE8B'  31  0172   448 90$:            BRW     IOC$UNLOCK              ;UNLOCK I/O DATA BASE AND RETURN
                          0175   449
                          0175   450 ;
                          0175   451 ; ASSIGNMENT OF A CLONED UCB
                          0175   452 ;
                          0175   453
                          0175   454 CLONE_UCB:
          0B 38 A5   0D   E1  0175   455                 BBC     #DEV$V_NET, -           ; Branch if this device is not a
                          017A   456                         UCB$L_DEVCHAR(R5), 210$ ; network device.
           50   28A4 8F   3C  017A   457                 MOVZWL  #SS$_NONETMBX, R0       ; Else, the process must have NETMBX
                          017F   458                 IFNPRIV NETMBX, 90$             ; privilege to perform this operation.
                FE78'  30  0185   459 210$:           BSBW    IOC$CHKUCBQUOTA         ; Does process have enough BYTLM quota?
              E7 50    E9  0188   460                 BLBC    R0, 90$                 ; Branch if insufficient BYTLM quota.
                FE72'  30  018B   461                 BSBW    IOC$CLONE_UCB           ; Make the clone UCB.
              E1 50    E9  018E   462                 BLBC    R0, 90$                 ; Branch if clone opeation failed.
                          0191   463
                          0191   464                 ASSUME ORB$L_OWNER EQ 0
                          0191   465
       1C B2    00BC C4   D0  0191   466                 MOVL    PCB$L_UIC(R4), -        ; Make the current UIC the owner of
                          0197   467                         @UCB$L_ORB(R2)          ; the cloned UCB.
          00 64 A2   10   E2  0197   468                 BBSS    #UCB$V_DELETEUCB, -     ; Mark the cloned UCB for automatic
                          019C   469                         UCB$L_STS(R2),213$      ; deletion when the ref. count reaches
                          019C   470                                                 ; zero.
          04 38 A2   14   E1  019C   471 213$:          BBC     #DEV$V_MBX, -           ; Does this device behave like a
                          01A1   472                         UCB$L_DEVCHAR(R2),215$  ; mailbox? Branch if not.
             68 A2   02   A8  01A1   473                 BISW    #UCB$M_DELMBX, -        ; Else, set mailbox-like delete bit.
                          01A5   474                         UCB$W_DEVSTS(R2)
                5C A2   B4  01A5   475 215$:          CLRW    UCB$W_REFC(R2)          ; Zero the cloned UCB reference count;
                          01A8   476                                                 ; it will be incremented when the
                          01A8   477                                                 ; channel assignment is completed.
                   FE55'  30  01A8   478                 BSBW    IOC$DEBIT_UCB           ; Debit process quota for cloned UCB.
           53   0088 C5   D0  01AB   479                 MOVL    UCB$L_DDT(R5), R3       ; Get DDT address.
                50   01   3C  01B0   480                 MOVZWL  #SS$_NORMAL, R0         ; Assume success return from driver.
                  24 B3   16  01B3   481                 JSB     @DDT$L_CLONEDUCB(R3)    ; Call the driver's cloned UCB routine.
                55   52   D0  01B6   482                 MOVL    R2, R5                  ; Make the cloned UCB the current UCB.
                   0D 50   E9  01B9   483                 BLBC    R0, 290$                ; Branch if driver vetos cloning.
       05 38 A5    10   E0  01BC   484                 BBS     #DEV$V_SHR, -           ; Branch if cloning a sharable UCB.
                          01C1   485                         UCB$L_DEVCHAR(R5), 240$
       2C A5    60 A4   D0  01C1   486                 MOVL    PCB$L_PID(R4), UCB$L_PID(R5) ; Else, do implicit allocation.
                   FF40   31  01C6   487 240$:           BRW     70$                     ; Go complete normal channel assignment.
                          01C9   488
                     50   DD  01C9   489 290$:          PUSHL   R0                      ; Save reason for aborting cloning oper.
                FE32'  30  01CB   490                 BSBW    IOC$CREDIT_UCB          ; Credit process quota for cloned UCB.
                FE2F'  30  01CE   491                 BSBW    IOC$DELETE_UCB          ; Delete cloned UCB.
                   50 8ED0  01D1   492                 POPL    R0                      ; Restore return status.
                   9C   11  01D4   493                 BRB     90$                     ; Complete operation with error status.
                          01D6   494
                          01D6   495                 .DSABL  LSB
```

```
                                                H 15
                     01D6   497          .SBTTL  REMOTE DEVICE SPECIFIED
                     01D6   498
                     01D6   499 NO_KRP: BUG_CHECK       KRPEMPTY,FATAL
                     01DA   500
                     01DA   501 ;
                     01DA   502 ; REMOTE DEVICE SPECIFIED
                     01DA   503 ;
                     01DA   504 ;       R9 STILL POINTS TO DEVNAM DESCRIPTOR
                     01DA   505 ;       R10 STILL POINTS TO ASSOCIATED MAILBOX DESCRIPTOR
                     01DA   506 REMOTE:
                     01DA   507
                     01DA   508 ;
                     01DA   509 ; CASE_BLIND FLAG (R5 INPUT) FOR LNM$SEARCH_ONE, CONCATENATE "USER-MODE"
                     01DA   510 ; FOR NOW
                     01DA   511 ;
00000103             01DA   512 M_CASE_BLIND    = ^X0103
                     01DA   513
                     01DA   514 ;
                     01DA   515 ; KRP USAGE FOR REMOTE LNM TRANSLATION
                     01DA   516 ;
00000000             01DA   517 LOGNAM          = 0                     ;LOGICAL NAME DESCRIPTOR
00000008             01DA   518 LNMX_OFFSET     = LOGNAM+8              ;LOGICAL NAME DATA
                     01DA   519
                     01DA   520 ;
                     01DA   521 ; SINCE THE KRP CONTAINS THE LOGICAL NAME EQUIVALENCE STRING AND THE
                     01DA   522 ; INPUTS TO THE $TRNLNM SERVICE, NEED TO MAKE SURE ONE KRP IS ENOUGH.
                     01DA   523 ;
                     01DA   524          ASSUME  <LNMX_OFFSET + <LNMX$T_XLATION+1> + LNM$C_NAMLENGTH> LE 512
                     01DA   525
          FE23'  30  01DA   526          BSBW    SCH$IOUNLOCK            ;UNLOCK I/O DATA BASE
                     01DD   527          SETIPL  #0                      ;ALLOW INTERRUPTS
53  00000000'GF  9E  01E0   528          MOVAB   G^CTL$GL_KRPFL,R3       ;GET ADDRESS OF KRP LOOKASIDE LIST
   58    04 B3  0F  01E7   529          REMQUE  @4(R3),R8               ;GET A KRP
          E9  1D  01EB   530          BVS     NO_KRP                  ;BUG_CHECK IF NO KRPS TO USE
      50  69  3C  01ED   531          MOVZWL  (R9),R0                 ;GET LENGTH OF LOGICAL NAME STRING
          6E  13  01F0   532          BEQL    60$                     ;BRANCH IF ILLEGAL LENGTH
                 01F2   533          ASSUME  LNM$C_NAMLENGTH-1 LE 512 ;ASSUME ONE PROBE WILL DO
  00FE 8F  50  B1  01F2   534          CMPW    R0,#LNM$C_NAMLENGTH-1   ;LOGICAL NAME STRING TOO LONG?
          67  14  01F7   535          BGTR    60$                     ;IF GTR, SIZE TOO LARGE
   51   04 A9  D0  01F9   536          MOVL    4(R9),R1                ;GET ADDRESS OF LOGICAL NAME STRING
                 01FD   537          IFNORD  R0,(R1),80$             ;PROBE INPUT LOGICAL NAME
                 0203   538 ;
                 0203   539 ; TRANSLATE THE NET LOGICAL NAME
                 0203   540 ;
      57  0A  D0  0203   541          MOVL    #LNM$C_MAXDEPTH,R7      ;10 TRANSLATION LIMIT
   56   08 A8  DE  0206   542          MOVAL   LNMX_OFFSET(R8),R6      ;OUTPUT BUFFER
          66  7C  020A   543          CLRQ    (R6)                    ;INITIALIZE LNMX
   04 A6  50  90  020C   544          MOVB    R0,LNMX$T_XLATION(R6)   ;INITIAL STRING LENGTH
          10  BB  0210   545          PUSHR   #^M<R4>                 ;SAVE PCB
      61  50  28  0212   546          MOVC3   R0,(R1),-               ;COPY INITIAL STRING TO LNMX
        05 A6  0215   547                  <LNMX$T_XLATION+1>(R6)    ;
          10  BA  0217   548          POPR    #^M<R4>                 ;RESTORE PCB
        0023  31  0219   549          BRW     30$                     ;TAKE FIRST STAB AT TRANSLATION
                 021C   550                                          ;LOGICAL NAME DESCRIPTOR IN R0/R1
52  FDE0 CF  3C  021C   551 10$:     MOVZWL  LNM_TBL,R2              ;GET TABLE NAME LENGTH
53  FDDF CF  D0  0221   552          MOVL    LNM_TBL+4,R3            ;AND TABLE NAME ADDRESS
55  0103 8F  3C  0226   553          MOVZWL  #M_CASE_BLIND,R5        ;CASE-BLIND TRANSLATION, USER-MODE
```

```
                00000000'EF   16   022B   554 20$:    JSB     LNM$SEARCH_ONE                  ;TRANSLATE LOGICAL NAME
                        0B 50 E8   0231   555         BLBS    R0,30$                          ;BRANCH IF TRANSLATION OCCURED
          50      01BC 8F   B1   0234   556         CMPW    #SS$_NOLOGNAM,R0                ;TRANSLATION FAILURE?
                        2A   12   0239   557         BNEQ    70$                             ;NO, SOME SERIOUS PROBLEM
                        01   E3   023B   558         BBCS    #LNMX$V_TERMINAL,-              ;INDICATE NO MORE TRANSLATIONS
                     00 66        023D   559                 LNMX$B_FLAGS(R6),30$            ;
                     05 A6   DE   023F   560 30$:    MOVAL   <LNMX$T_XLATION+1>(R6),-        ;RESET LOGICAL NAME DESCRIPTOR
                        51        0242   561                 R1                              ; ADDRESS
          50         04 A6   9A   0243   562         MOVZBL  LNMX$T_XLATION(R6),R0           ; SIZE
          61         5F 8F   91   0247   563 40$:    CMPB    #^A/_/,(R1)                     ;TRANSLATED NAME START WITH UNDERSCORE?
                        0A   12   024B   564         BNEQ    50$                             ;IF NEQ NO
                        50   D7   024D   565         DECL    R0                              ;DECREMENT LENGTH OF TRANSLATED NAME
                        0F   13   024F   566         BEQL    60$                             ;BRANCH IF LENGTH ILLEGAL
                        51   D6   0251   567         INCL    R1                              ;INCREMENT STARTING ADDRESS OF NAME
                        01   E3   0253   568         BBCS    #LNMX$V_TERMINAL,-              ;TERMINAL DUE TO PRESENCE OF "_",
                     F0 66        0255   569                 LNMX$B_FLAGS(R6),40$            ; LOOK FOR ANOTHER "_"
                        01   E0   0257   570 50$:    BBS     #LNMX$V_TERMINAL,-              ;IF DONE, GO ASSIGN CHANNEL
                     11 66        0259   571                 LNMX$B_FLAGS(R6),90$            ;
                     BE 57   F4   025B   572         SOBGEQ  R7,10$                          ;KEEP ON TRANSLATING
                        61   11   025E   573         BRB     100$                            ;OOPS, TOO MANY TRANSLATIONS
          50      0154 8F   3C   0260   574 60$:    MOVZWL  #SS$_IVLOGNAM,R0                ;INPUT SIZE TO LARGE
                        70   11   0265   575 70$:    BRB     120$                            ;
                  50    0C   3C   0267   576 80$:    MOVZWL  #SS$_ACCVIO,R0                  ;CAN NOT READ WHERE DESC POINTS
                        6B   11   026A   577         BRB     120$                            ;
                  68    50   7D   026C   578 90$:    MOVQ    R0,LOGNAM(R8)                   ;SAVE DESCRIPTOR IN A BETTER PLACE
                  59    68   DE   026F   579         MOVAL   LOGNAM(R8),R9                   ;SET ADDRESS OF TRANSLATED NAME DESCRIPTOR
                  FD9E  CF   9F   0272   580         PUSHAB  NETNAM                          ;BUILD NETWORK DEVICE NAME DESCRIPTOR
                        04   DD   0276   581         PUSHL   #NETEND-NETNAM                  ;
                  56    5E   D0   0278   582         MOVL    SP,R6                           ;SAVE ADDRESS OF NAME STRING DESCRIPTOR
                  57    7E   DE   027B   583         MOVAL   -(SP),R7                        ;ALLOCATE SPACE TO STORE CHANNEL NUMBER
                             7E   027E   584         $ASSIGN_S (R6),(R7),MAXACMODE(FP),(R10) ;ASSIGN CHANNEL TO NETWORK
                     46 50   E9   028E   585         BLBC    R0,120$                         ;IF LBC ASSIGNMENT FAILURE
                             91   0291   586         $QIOW_S S^#EXE$C_SYSEFN,(R7),#IO$_ACCESS!IO$M_ACCESS,,(R6),,,,R9
                             AE   02AE   587                                                 ;CONNECT TO NETWORK
                     17 50   E9   02AE   588         BLBC    R0,110$                         ;IF LBC SERVICE FAILURE
                  50    66   3C   02B1   589         MOVZWL  (R6),R0                         ;GET I/O COMPLETION CODE
                     11 50   E9   02B4   590         BLBC    R0,110$                         ;IF LBC CONNECT FAILURE
                  6B    67   B0   02B7   591         MOVW    (R7),(R11)                      ;STORE ASSIGNED DEVICE CHANNEL NUMBER
          50      0649 8F   3C   02BA   592         MOVZWL  #SS$_REMOTE,R0                  ;SET COMPLETION STATUS
                        16   11   02BF   593         BRB     120$                            ;
          50      0374 8F   3C   02C1   594 100$:    MOVZWL  #SS$_TOOMANYLNAM,R0             ;TOO MANY EQUIVALENCE NAMES DEFINED
                        0F   11   02C6   595         BRB     120$                            ;
                        50   DD   02C8   596 110$:    PUSHL   R0                              ;SAVE FINAL STATUS
                             CA   02CA   597         $DASSGN_S (R7)                          ;DEASSIGN CHANNEL
                        50 8ED0   02D4   598         POPL    R0                              ;RETRIEVE FINAL STATUS
                             D7   02D7   599 120$:                                           ;ONLY REMAINING WORK - RETURN KRP
                             D7   02D7   600                                                 ;R8 STILL POINTS TO KRP
          53    00000000'GF   9E   02D7   601         MOVAB   G^CTL$GL_KRPFL,R3               ;GET ADDRESS OF KRP LOOKASIDE LIST
                  04 B3   68   0E   02DE   602         INSQUE  (R8),@4(R3)                     ;RETURN KRP TO LIST
                        04   02E2   603         RET                                         ;
```

```
                                                J 15
                     02E3    605          .SBTTL  TEST IF MAILBOX SPECIFIED
                     02E3    606 ;
                     02E3    607 ; SUBROUTINE TO TEST IF A MAILBOX IS SPECIFIED
                     02E3    608 ;
                     02E3    609 ; INPUTS:
                     02E3    610 ;
                     02E3    611 ;       R10 = ADDRESS OF MAILBOX NAME DESCRIPTOR
                     02E3    612 ;
                     02E3    613 ; OUTPUTS:
                     02E3    614 ;
                     02E3    615 ;       R0 = SS$_NORMAL IF SPECIFIED MAILBOX EXISTS
                     02E3    616 ;            SS$_NOSUCHDEV IF SPECIFIED MAILBOX DOES NOT EXIST
                     02E3    617 ;            SS$_DEVNOTMBX IF SPECIFIED DEVICE IS NOT A MAILBOX
                     02E3    618 ;       R6 = ADDRESS OF MAILBOX UCB
                     02E3    619 ;            ZERO IF MAILBOX NOT SPECIFIED (USED AS CHANNEL STATUS FLAGS)
                     02E3    620 ;
                     02E3    621
                     02E3    622 TEST_MAILBOX:                                   ;
          56   5A    D0 02E3 623          MOVL    R10,R6                         ;SET ADDRESS OF MAILBOX NAME DESCRIPTOR
               1B    13 02E6 624          BEQL    10$                            ;IF EQL NO NAME SPECIFIED
          51   56    D0 02E8 625          MOVL    R6,R1                          ;COPY ADDRESS OF MAILBOX NAME DESCRIPTOR
                FD12' 30 02EB 626          BSBW    IOC$SEARCHDEV                  ;SEARCH FOR DEVICE
              15 50   E9 02EE 627          BLBC    R0,20$                         ;IF LBC SEARCH ERROR
     50   0074 8F    3C 02F1 628          MOVZWL  #SS$_DEVNOTMBX,R0              ;SET DEVICE NOT MAILBOX STATUS
  0B 38   A1   14    E1 02F6 629          BBC     S^#DEV$V_MBX,UCB$L_DEVCHAR(R1),20$ ;IF CLR, DEVICE NOT MAILBOX
  06 38   A1   0D    E0 02FB 630          BBS     S^#DEV$V_NET,UCB$L_DEVCHAR(R1),20$ ;IF SET, NETWORK DEVICE
          56   51    D0 0300 631          MOVL    R1,R6                          ;SAVE ADDRESS OF MAILBOX UCB
          50   01    3C 0303 632 10$:     MOVZWL  #SS$_NORMAL,R0                 ;SET NORMAL COMPLETION STATUS
                     05 0306 633 20$:     RSB                                    ;
                     0307 634
                     0307 635          .END
```

```
$$T1                    = 00000001
ACMODE                  = 0000000C
BUG$_KRPEMPTY             ********   X   02
CCB$B_AMOD              = 00000009
CCB$B_STS               = 00000008
CCB$L_UCB               = 00000000
CCB$M_AMB               = 00000001
CHAN                    = 00000008
CLONE_UCB                 00000175 R     02
CLU$GL_CLUB               ********   X   02
CLUB$L_FLAGS            = 0000001C
CLUB$V_CLUSTER          = 00000000
CTL$GL_KRPFL              ********   X   02
DDT$L_CLONEDUCB         = 00000024
DEV$V_CLU               = 00000000
DEV$V_FOD               = 0000000E
DEV$V_MBX               = 00000014
DEV$V_NET               = 0000000D
DEV$V_RED               = 00000008
DEV$V_SHR               = 00000010
DEV$V_SPL               = 00000006
DEV$V_SSM               = 00000006
DEVNAM                  = 00000004
DIR...                  = FFFFFFFF
EXE$ASSIGN                00000018 RG    02
EXE$CHKRDACCES            ********   X   02
EXE$C_SYSEFN              ********   X   02
EXE$MAXACMODE             ********   X   02
FREECHAN                  00000066 R     02
IO$M_ACCESS             = 00000040
IO$_ACCESS              = 00000032
IOC$CHKUCBQUOTA           ********   X   02
IOC$CLONE_UCB             ********   X   02
IOC$CREDIT_UCB            ********   X   02
IOC$DEBIT_UCB             ********   X   02
IOC$DELETE_UCB            ********   X   02
IOC$FFCHAN                ********   X   02
IOC$LOCK_DEV              ********   X   02
IOC$M_ANY               = 00000040
IOC$M_PHY               = 00000001
IOC$SEARCH                ********   X   02
IOC$SEARCHDEV             ********   X   02
IOC$UNLOCK                ********   X   02
IOC$V_NO_TRANS          = 00000009
LCK$K_CRMODE            = 00000001
LNM$C_MAXDEPTH          = 0000000A
LNM$C_NAMLENGTH         = 000000FF
LNM$SEARCH_ONE            ********   X   02
LNMX$B_FLAGS            = 00000000
LNMX$T_XLATION          = 00000004
LNMX$V_TERMINAL         = 00000001
LNMX_OFFSET             = 00000008
LNM_TBL                   00000000 R     02
LOCAL                     000000BA R     02
LOGNAM                  = 00000000
MAXACMODE                 FFFFFFFC
MBXNAM                  = 00000010
M_CASE_BLIND            = 00000103
NETEND                    00000018 R     02
NETNAM                    00000014 R     02
NO_KRP                    000001D6 R     02
ORB$L_OWNER             = 00000000
PCB$L_OWNER             = 0000001C
PCB$L_PID               = 00000060
PCB$L_UIC               = 000000BC
PCB$Q_PRIV              = 00000084
PR$_IPL                 = 00000012
PRV$V_NETMBX            = 00000014
PRV$V_PHY_IO            = 00000016
PRV$V_SHARE             = 0000001F
REMOTE                    000001DA R     02
SAVABS...               = FFFFFFFC
SCH$GL_PCBVEC             ********   X   02
SCH$IOLOCKW               ********   X   02
SCH$IOUNLOCK              ********   X   02
SS$_ACCVIO              = 0000000C
SS$_DEVACTIVE           = 000002C4
SS$_DEVALLOC            = 00000840
SS$_DEVNOTMBX           = 00000074
SS$_IVDEVNAM            = 00000144
SS$_IVLOGNAM            = 00000154
SS$_NOLOGNAM            = 000001BC
SS$_NONETMBX            = 000028A4
SS$_NONLOCAL            = 000008F0
SS$_NOPHY_IO            = 000028B4
SS$_NORMAL              = 00000001
SS$_REMOTE              = 00000649
SS$_TOOMANYLNAM         = 00000374
SYS$ASSIGN                ********  GX   02
SYS$DASSGN                ********  GX   02
SYS$QIOW                  ********  GX   02
TEST_MAILBOX              000002E3 R     02
UCB$L_AMB               = 00000060
UCB$L_DDT               = 00000088
UCB$L_DEVCHAR           = 00000038
UCB$L_DEVCHAR2          = 0000003C
UCB$L_ORB               = 0000001C
UCB$L_PID               = 0000002C
UCB$L_STS               = 00000064
UCB$L_TT_LOGUCB         = 000000C0
UCB$M_DELMBX            = 00000002
UCB$V_DELETEUCB         = 00000010
UCB$V_TEMPLATE          = 0000000D
UCB$W_DEVSTS            = 00000068
UCB$W_REFC              = 0000005C
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	FFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
Y$EXEPAGED	00000307 (775.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:02.23
Command processing	110	00:00:00.58	00:00:05.50
Pass 1	524	00:00:21.47	00:01:06.29
Symbol table sort	0	00:00:03.69	00:00:10.88
Pass 2	127	00:00:03.69	00:00:11.55
Symbol table output	13	00:00:00.11	00:00:00.64
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	807	00:00:29.63	00:01:37.13

The working set limit was 1800 pages.
122209 bytes (239 pages) of virtual memory were used to buffer the intermediate code.
There were 130 pages of symbol table space allocated to hold 2355 non-local and 40 local symbols.
635 source lines were read in Pass 1, producing 18 object records in Pass 2.
44 pages of virtual memory were used to define 42 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	19
_255DUA28:[SYSLIB]STARLET.MLB;2	20
TOTALS (all libraries)	39

2617 GETS were required to define 39 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SYSASSIGN/OBJ=OBJ$:SYSASSIGN MSRC$:SYSASSIGN/UPDATE=(ENH$:SYSASSIGN)+EXECML$/LIB
```

0381 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SSDEF
LIS

SYSACPFDT
LIS

SYSADJSTK
LIS

SYSASCEFC
LIS

SYSBRKTHR
LIS

SYSASSIGN
LIS

SWAPPER
LIS

SYSASTCON
LIS

SVAPTE
LIS

SYSADJWSL
LIS